Title:
Improved Collision-Free Multi-Axis Tool-Path for Additive Manufacturing

Authors:
Rahnuma Islam Nishat, rahnuma.nishat@ubc.ca, University of British Columbia, Canada
Yeganeh Bahoo, bahoo@torontomu.ca, Toronto Metropolitan University, Canada
Konstantinos Georgiou, konstantinos@torontomu.ca, Toronto Metropolitan University, Canada
Robert Hedrick, bob.hedrick@camufacturing.com, CAMufacturing Solutions Inc., Canada
R. Jill Urbanic, jurbanic@uwindsor.ca, University of Windsor, Canada

Keywords:
Collision Avoidance, Mesh, Shortest Path, Graph Theory, Heuristic Methods

DOI: 10.14733/cadconfP.2023.339-343

Introduction:

Additive manufacturing (AM) is a new way of manufacturing in which materials are fabricated layer by layer, increasing the volume of the object being printed over the course of manufacturing. The increase in the volume causes higher chances of collision between the machine parts and the printed material during the fabrication process. In many additive manufacturing system, a planar '$2\frac{1}{2}$D' build strategy is employed where collision avoidance issues do not exist between the object being printed and the AM heat source solution (e.g., a laser or a a material deposition nozzle). However, the directed energy deposition (DED) processes use a heat source and material feeding system mounted on a multi-axis CNC system or a robot to deposit beads side by side to fill a layer. The DED processes are getting popular as they can be used to fabricate parts without support structure, repair damaged parts, or in surface coating, by leveraging non-planar slicing and multi-axis tool-paths. This multi-axis solution introduces potential collision issues.

The state-of-the-art algorithms for designing tool-paths for multi-axis additive manufacturing do not consider the above collision issues and thus, have no techniques to avoid such collisions. However, such collisions does not only affect the printing process and the quality of the printing, but can also incur major damages to the machines themselves. Therefore, addressing this issues is of utmost importance and a matter of urgency . We need to put special consideration in the design of the *tool-path*, which is the path on which the *deposit head* of the machine moves on during the manufacturing process. A *tool-path* is a path defined by a sequence of points in 3D along with *tool vector* for each tool-path point which is a vector representing the direction of the deposit head at that point, and difference between tool vectors of two consecutive points of the tool-path does not exceed some given threshold.

Although several approaches for detecting and avoiding collisions have been investigated for CNC machining (see [10] for a survey), it is a relatively new research area for AM processes. The AM problem space has unique challenges to overcome. First of all, the deposit head may not be rotationally symmetric as shown in Figure 1(a)–(b), where the cutting tools in machining are symmetric. Secondly, as mentioned above, AM process adds material to the workspace in contrast to subtractive manufacturing or machining where materials are removed over time, thus increasing the probability of collision with the progress of the building process. Finally, the build shape (see Figure 1(c)–(d) for some example) cannot be accurately represented due to the basic imprecision of the DED AM process.

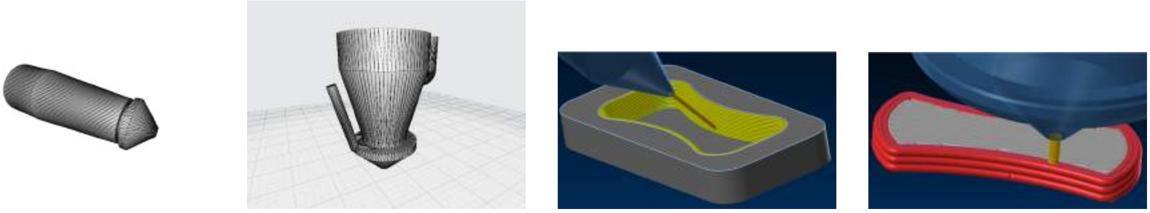

Fig. 1: From left to right: (a) rotationally symmetric holder, and (b) rotationally asymmetric holder; (c) printing inside a cavity where the deposit head may collide with the object; (d) printing boundary of an object.

Previously, we gave an algorithm that modifies a given multi-axis AM tool-path to a collision-free tool-path [3, 4]. Using 3D triangle-mesh representations for both the object to be printed and the deposit head, we first applied algorithms from the literature to detect collision between those two meshes. If collision is detected, we built a *configuration graph* considering multiple tool vectors for each tool-path point as vertices and computed a collision-free tool-path, if it exists, using graph algorithms to compute a path in the configuration graph containing a collision-free tool vector for each tool-path point. Although the algorithm showed promising results, the time required by the algorithm for complicated and big tool-paths was not up to the standard in the AM industry. In this paper, we propose heuristic based approaches to improve the performance of the previous algorithm significantly.

The rest of the paper is organized as follows. We discuss some related work in the next section and also give an overview of our previous algorithm from [3, 4]. We then describe our proposed heuristic approach. Finally, we conclude the paper with a discussion on future direction of research in avoiding collision in multi-axis additive manufacturing tool-path generation.

Related Work:

Collision avoidance in CNC machining has been studied for both *local* (i.e., local gouging, rear gouging, etc.) and *global* collisions. Balasubramanium et al. [6] gave a three-step algorithm to generate tool-path using visibility and accessibility-based method. However, their algorithm is computationally expensive as the objects being machined are presented as *clouds of points*. Wang and Tang [11] proposed an algorithm to identify the set of valid orientations or configurations by inspecting the valid area with all the gouging constraints. But their method had a high space and time complexity as well.

Collision detection and avoidance has attracted attention in additive manufacturing as well. Nishat et al. [3, 4] gave an algorithm to obtain collision-free tool-paths for multi-axis additive manufacturing (see the next section for details on the algorithm). Fang et al. [5] studied collision-free printing of a layer by selecting a 'best' setup orientation, similar to choosing collision-free tool vectors in our algorithm from [3, 4]. Plakhotnik et al. [1] studied collision avoidance by updating the tool vectors in multi-axis additive manufacturing. They penalized the tool vectors as they get further from the position normal to the surface. However, our algorithm [3, 4] penalized bigger differences between tool vectors of adjacent tool-path points. Jiang et al. [2] studied scheduling in path planning to avoid collision in additive manufacturing when multiple deposit heads are employed.

In computational geometry, detecting intersection between geometric objects has been studied for decades. Möller [9] and Held [8] studied detecting intersection between 3D triangles. Guigue and Devillers [7] improved the above algorithms using floating-point calculation.

Proceedings of CAD'23, Mexico City, Mexico, July 10-12, 2023, 339-343

Computing Collision-Free Tool-Path:

In this section, we give a summary of our previous algorithm [3, 4] to generate collision-free tool-path. We propose algorithms to improve this approach in the next section.

Given a tool-path, which includes a set of continuous positions of the deposit head a long with tool vector of each points (direction of the deposit head), our goal is to find a collision-free tool-path if it exist for this set of points. We present both the surface of the CAD model and the deposit head as triangular mesh. We considered both symmetric and asymmetric deposit heads; see Figure 1(a) and (b), respectively, for symmetric and asymmetric deposit head mesh. An example run of the algorithm is shown in Figure 2, the tool-paths were generated using APlus software on Mastercam platform.

Fig. 2: [3, 4] Generating collision-free tool-path from a given tool-path. From left to right: (a) tool-path with 566 points creates collision between the printed surface and the asymmetric deposit head; (e) original tool vectors for (a); (f) Modified tool-path that avoids collision.

We applied two geometric approaches to detect collisions between the deposit head and the object being printed: 1) Möller's algorithm [9], and 2) a commercial third-party library. The first was based on calculation of collisions between each pairs of triangles of the mesh of the object and the mesh of the deposit head. To speed up the Möller's technique we used the parallel computation. The second approach was based on clash detection between two solids representing the deposit head and the object; multi-threading was used to apply this technique.

If collision was detected, our algorithm modified the given tool-path to a collision-free tool-path. We built a *configuration graph* where each vertex represents a tool vector at a specific tool-path point. A set of *feasible* tool vectors was assigned to each point of the tool-path, where a *feasible* tool vector means that the deposit head does not collide with the object when aligned along this tool vector at that specific tool-path point. Two vertices will be connected by an *edge* in the configuration graph if and only if their representative tool vectors are for two consecutive points on the tool-path, and the deposit head can change its direction by the difference between the angles of this two tool vector, which is determined based on the mechanical configuration of the machine. This connection is directional, connecting the vertices of the tool-path position x to y when x appears before y. We created a start point s and connect s to all the vertices allocated to the first point of the tool-path. We also created the vertex t and connected all the vertices associated to the last point of the tool-path to t.

After creating the above directed configuration graph, we ran the breadth-first search or Dijkstra's shortest path algorithm to find a path from s and t through all the tool-path points. If such a path exists, that represents a collision-free tool-path for the deposit head. Otherwise, no collision-free tool-path exist for the deposit head to print on the given path. Applying Dijkstra's algorithm gave us the path that allowed minimum change of angles between tool vectors of consecutive tool-path points.

The configuration graph in our algorithm [3, 4] was large even for tool-paths which includes a few hundreds of points. This slowed down the computation of the path from s to t. To improve the time complexity, we have studied different ways of approaches including heuristic approach to improve the time. We will go over the details of our approach in the next section.

Heuristic Approaches:

In order to reduce the time we need to calculate the collision-free tool-path, we reduced the size of the produced graph using the following two heuristic approaches; see Figure 3 for the flowcharts. Note that the heuristic approaches can be applied separately, or together to improve the running time.

Heuristic 1. This approach reduces the number of feasible tool vector considered for each tool-path point. Instead of considering a number of tool vectors for all the tool-path points, we consider multiple feasible tool vectors for only the tool-path points where the deposit head collides with the object. For each such point, we first consider feasible tool vectors closest to the original tool vector and create the configuration graph. If a collision-free tool-path is generated from the configuration graph, we return it. Otherwise, we consider more feasible tool vectors for the collision points that are farther from the original tool vector, and instead of creating the configuration again we add the new vertices and respective edges to the already created configuration graph. We then run the algorithm to find a path from s to t again. In this way, we keep on adding more feasible tool vectors as vertices of the configuration graph, until we find a collision-free tool-path or no more feasible tool vectors remain for the collision points. In the later case, if no collision-free path is found at that point, the algorithm declares that it could not find a collision-free path.

Heuristic 2. This approach reduces the size of the configuration graph by reducing the number of tool-path points considered. Instead of including the entire tool-path, we sample every tenth point for the graph, the difference between tool vectors angles are adjusted accordingly. If a collision-free tool-path is found, we then move into each interval and find the collision-free tool-path for each interval. If the tool-path in all intervals abide by the mechanical specification of the AM machine, we report the constructed path as the final result.

Conclusion:

In this work, we improved the efficiency of the fabrication of the collision-free tool-path by using heuristic methods. This was achieved by focusing on the points of the path in which collision occurs, as well as reducing the overall size of the configuration graph. Additionally, using the heuristic approach A* instead of Dijkstra's algorithm also helps efficiency.

References:

[1] Plakhotnik D.; Glasmacher L.; Vaneker T.; Smetanin Y.; Stautner M.; Murtezaoglu Y.; van Houten F.: CAM planning for multi-axis laser additive manufacturing considering collisions. CIRP Annals, 68(1), 2019, 447–450. https://doi.org/10.1016/j.cirp.2019.04.007

[2] Jiang Z.; Wang H.; Sun Y.: Improved co-scheduling of multi-layer printing path scanning for collaborative additive manufacturing. IISE Transactions, 53(9), 2021, 960–973. https://doi.org/10.1080/24725854.2020.1807076

[3] Nishat R.I.; Bahoo Y.; Georgiou K.; Hedrick R.; Urbanic R.J.: Collision-Free Multi-Axis Tool-Path for Additive Manufacturing Proceedings of CAD'22, Beijing, China, July 11-13, 2022, 297–301. https://doi:10.14733/cadconfP.2022.297-301

[4] Nishat R.I.; Bahoo Y.; Georgiou K.; Hedrick R.; Urbanic R.J.: Collision-Free Multi-Axis Tool-Path for Additive Manufacturing Computer-Aided Design and Applications, 20(6), 1094-1109, 2023. http://doi.org/10.14733/cadaps.2023.1094-1109

[5] Fang G.; Zhang T.; Zhong S.; Chen X.; Zhong Z. Wang C.: Reinforced FDM: multi-axis filament alignment with controlled anisotropic strength. ACM Transactions on Graphics, 39(6), 2020, Article 204, 1–15. https://doi.org/10.1145/3414685.3417834

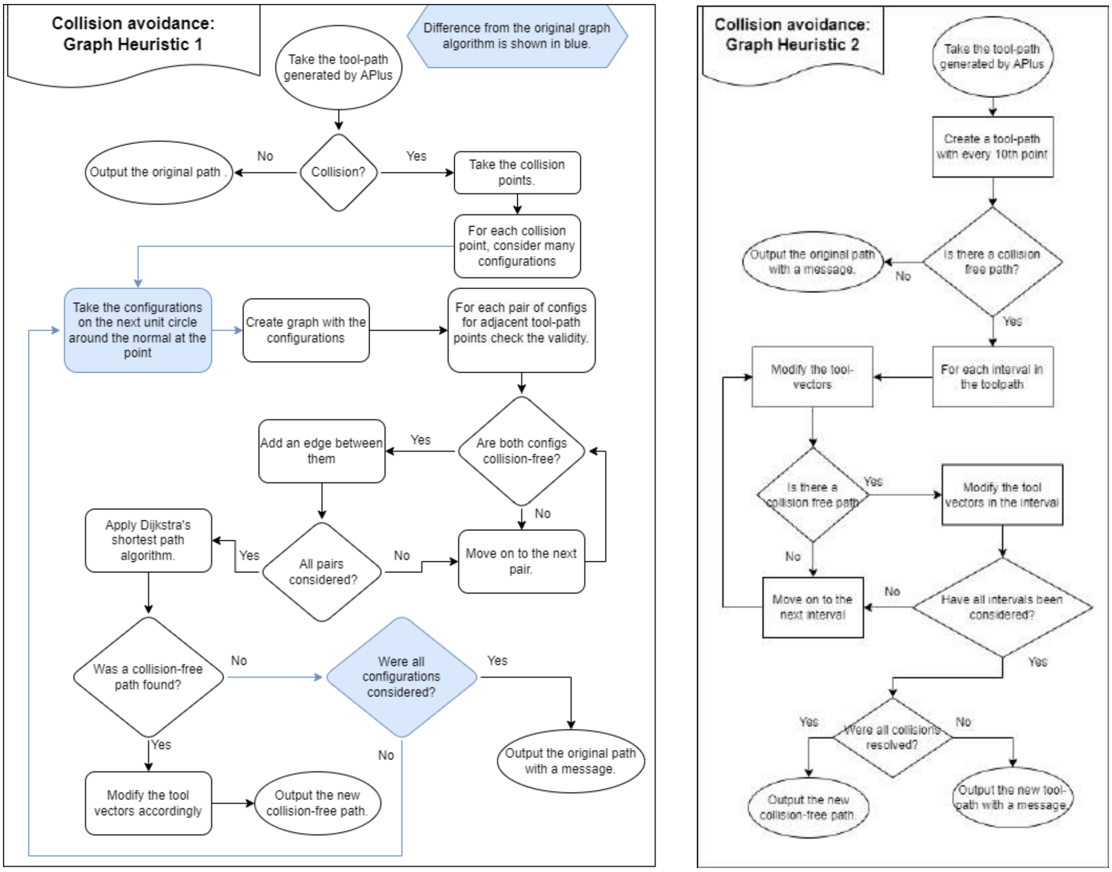

Fig. 3: The flow charts of the heuristic approaches 1 and 2.

[6] Balasubramaniam M.;Laxmiprasad P.; Sarma S.; Shaikh Z.: Generating 5-axis NC roughing paths directly from a tessellated representation. Computer-Aided Design, 32(4), 2000, 261–277. https://doi.org/10.1016/S0010-4485(99)00103-7

[7] Devillers O.; Guigue P.: Faster Triangle-Triangle Intersection Tests. Technical Report RR-4488, INRIA, June 2002. https://hal.inria.fr/inria-00072100

[8] Held M.: Erit - a collection of efficient and reliable intersection tests. Journal of Graphics Tools, 2, 1998, 25–44. https://doi.org/10.1080/10867651.1997.10487482

[9] Möller T.: A fast triangle-triangle intersection test. Journal of Graphics Tools, 2, 1997, 25–30. https://doi.org/10.1080/10867651.1997.10487472

[10] Tang T.D.: Algorithms for collision detection and avoidance for five-axis NC machining: a state of the art review. Computer-Aided Design, 51, 2014, 1–17. https://doi.org/10.1016/j.cad.2014.02.001

[11] Wang N.; Tang K.: Automatic generation of gouge-free and angular-velocity-compliant five-axis toolpath. Computer-Aided Design, 39(10), 2007, 841–852. https://doi.org/10.1016/j.cad.2007.04.003

Proceedings of CAD'23, Mexico City, Mexico, July 10-12, 2023, 339-343